используется для обеспечения конфиденциальности передаваемой по радиоканалу информации в стандарте мобильной сотовой связи GSM. A8 является одним из алгоритмов обеспечения секретности разговора в GSM вместе с A5 и A3. Его задача — генерация сеансового ключа Kc для потокового шифрования информации в канале связи между сотовым телефоном (MS — Mobile Station) и базовой станцией (BTS — Basic Transmitter Station) после аутентификации. По причине безопасности формирование Kc происходит в Sim-карте.

Для предотвращения несанкционированного использования системы связи используются механизмы аутентификации. Каждый подвижный абонент имеет стандартный модуль аутентификации (SIM-карту), который содержит международный идентификационный номер мобильного абонента (IMSI), индивидуальный 128-битный ключ аутентификации (Ki), а также алгоритмы аутентификации (A3) и генерации сеансового ключа (A8).

Ключ аутентификации пользователя Ki уникален и связан с IMSI. Оператор связи может определить Ki по значению IMSI. От несанкционированного использования SIM-карта защищена PIN-кодом, который предоставляется пользователю вместе с картой.

Процедура проверки подлинности абонента выглядит следующим образом. Сеть генерирует случайный номер (RAND), который передается на мобильное устройство. В SIM-карте происходит вычисление значения отклика (SRES) и сеансового ключа с использованием RAND, Ki, и алгоритмов A3, A8. Мобильное устройство вычисляет значение SRES и отправляет его в сеть, где оно сравнивается с вычисленным значением

RC4

RC4 (от англ. Rivest cipher 4 или Ron’s code), также известен как ARC4 или ARCFOUR (alleged RC4) — потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и TLS, алгоритмах обеспечения безопасности беспроводных сетей WEP и WPA).

Шифр разработан компанией RSA Security, и для его использования требуется лицензия.

Основные преимущества шифра:

− высокая скорость работы;

− переменный размер ключа.

− RC4 довольно уязвим, если:

− используются не случайные или связанные ключи;

− один ключевой поток используется дважды.

Алгоритм шифрования:

1) функция генерирует последовательность битов (ki);

2) затем последовательность битов посредством операции «суммирование по модулю два» (xor) объединяется с открытым текстом (si). В результате получается шифрограмма (ci): ci=si XOR ki.

Схема работы алгоритма RC4 представлена в таблице 40.

Алгоритм расшифровки.

1) повторно создаётся (регенерируется) поток битов ключа (ключевой поток) (ki).

2) поток битов ключа складывается с шифрограммой (ci) операцией «xor». В силу свойств операции «xor» на выходе получается исходный (незашифрованный) текст (si): si=ci XOR ki = (si XOR ki) XOR ki

Таблица 40. Шифрование

Незашифрованное сообщение	s1	s2	s3	s4	s5	…	si
	XOR						
Ключ	k1	k2	k3	k4	k5	…	ki
				⇩			
Зашифрованное сообщение	c1	c2	c3	c4	c5	…	ci

Таблица 41. Дешифрование

Зашифрованное сообщение	c1	c2	c3	c4	c5	…	ci
	XOR						
Ключ	k1	k2	k3	k4	k5	…	ki
				⇩			
Незашифрованное сообщение	s1	s2	s3	s4	s5	…	si

RC4 — фактически класс алгоритмов, определяемых размером блока (в дальнейшем S-блока). Параметр *n* является размером слова для алгоритма и определяет длину *S-блока*. Обычно, *n* = 8, но в целях анализа можно уменьшить его. Однако для повышения безопасности необходимо увеличить эту величину. В алгоритме нет противоречий на увеличение размера S-блока. При увеличении n, допустим, до 16 бит, элементов в S-блоке становится 65 536 и соответственно время начальной итерации будет увеличено. Однако, скорость шифрования возрастёт.

Внутреннее состояние RC4 представляется в виде массива размером 2^n и двух счётчиков. Массив известен как S-блок, и далее будет обозначаться как S. Он всегда содержит перестановку 2^n возможных значений слова. Два счётчика обозначены через i и j.

Инициализация RC4 состоит из двух частей:

1) инициализация S-блока;

2) генерация псевдослучайного слова K.

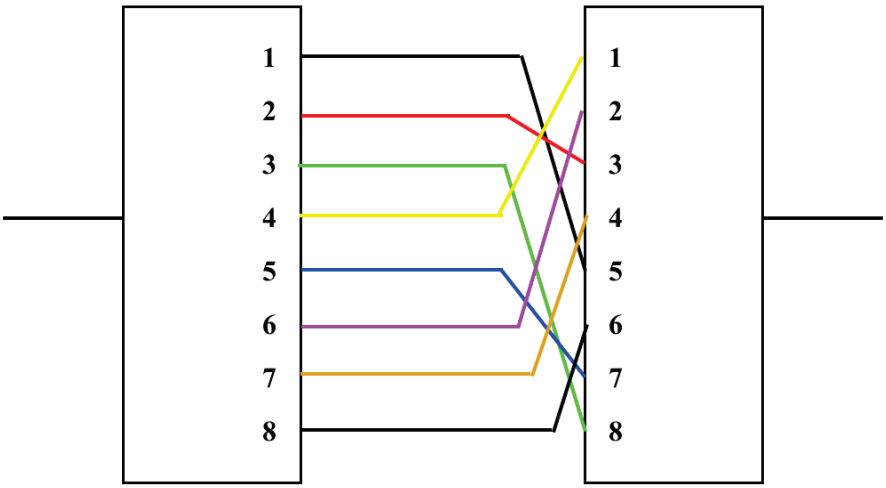

Рисунок 52. Схема S-блока

Пример

Сообщение: МИРЭА

Ключ: КНИГА

Таблица 42. Двоичный код букв

№	Буква	Двоичный код	№	Буква	Двоичный код	№	Буква	Двоичный код
1	А	00000001	12	К	00001100	23	Х	00010111
2	Б	00000010	13	Л	00001101	24	Ц	00011000
3	В	00000011	14	М	00001110	25	Ч	00011001
4	Г	00000100	15	Н	00001111	26	Ш	00011010

Окончание табл. 42

№	Буква	Двоичный код	№	Буква	Двоичный код	№	Буква	Двоичный код
5	Д	00000101	16	О	00010000	27	Щ	00011011
6	Е	00000110	17	П	00010001	28	Ъ	00011100
7	Ё	00000111	18	Р	00010010	29	Ы	00011101
8	Ж	00001000	19	С	00010011	30	Ь	00011110
9	З	00001001	20	Т	00010100	31	Э	00011111
10	И	00001010	21	У	00010101	32	Ю	00100000
11	Й	00001011	22	Ф	00010110	33	Я	00100001

Таблица 43. Пример шифрования

	3	2	4	5	1
Сообщение	М	И	Р	Э	А
	00001101	00001001	00010001	00011110	00000001
S-блок	8	3	1	7	5
Ключ	К	Н	И	Г	А
	00001011	00001110	00001001	00000100	00000001
XOR	00000110	00000111	00011000	00011010	00000000
Зашифрованное сообщение	Е	Ж	Ч	Щ	_

Ответ: 83175, ЕЖЧЩ_

Задача: расшифровать сообщение и получить комбинацию с помощью S-блока.

Вариант 1

Ключ	КИБ
Зашифрованное сообщение	ВОЪШЧХ ОЧХЪШВ
Комбинация S-блока	372815

Вариант 2

Ключ	КТСО
Зашифрованное сообщение	СПФЕ_ДА СПЕДФА
Комбинация S-блока	7531284

Вариант 3

Ключ	КИИБ
Зашифрованное сообщение	ЗЗНГ_ЧЫ _ЗЫГНЧЗ
Комбинация S-блока	7541823

Вариант 4

Ключ	ШИФР
Зашифрованное сообщение	ЗОЧЧЦПК КОЧЧЦЗП
Комбинация S-блока	4318752

Вариант 5

Ключ	КЛЮЧ
Зашифрованное сообщение	ЙБУБЭЛ БЙУЭБЛ
Комбинация S-блока	158732

Вариант 6

Ключ	КИЙ
Зашифрованное сообщение	ЗЗ_ _ЪЧ
Комбинация S-блока	857132

Вариант 7

Ключ	ОМ
Зашифрованное сообщение	ЛНМЕ_А_Э
Комбинация S-блока	85214736

Вариант 8

Ключ	БИТ
Зашифрованное сообщение	Е_ЬТРУЛ
Комбинация S-блока	7241385

Вариант 9

Ключ	ФАЙЛ
Зашифрованное сообщение	ХЗЕНОСТ
Комбинация S-блока	5382471

Вариант 10

Ключ	ПУХ
Зашифрованное сообщение	ПЬЬ_ДБД
Комбинация S-блока	1754382

SEAL только теория

SEAL (англ. Software-optimized Encryption Algorithm, программно-оптимизированный алгоритм шифрования) — симметричный поточный алгоритм шифрования данных, оптимизированный для программной реализации.

Разработан в IBM Филом Рогэвеем и Доном Копперсмитом в 1993 году. Алгоритм оптимизирован и рекомендован для 32-битных процессоров. Для работы ему требуется кэш-память на несколько килобайт и восемь 32-битовых регистров.

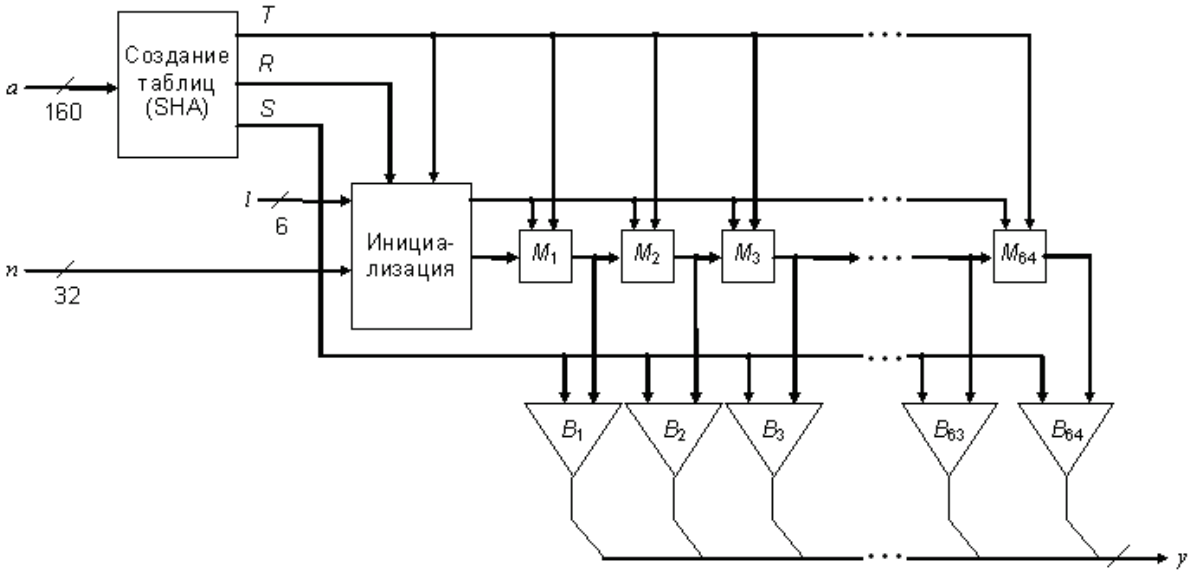

Рисунок 53. Схема алгоритма SEAL

Чтобы избежать потери скорости шифрования на медленных операциях алгоритм использует три таблицы: R, S и T. Эти таблицы вычисляются с помощью процедуры из алгоритма SHA-1 и зависят только от ключа. Заполнение данных таблиц можно описать с помощью функции G, которая из 160-битной строки и 32-битного числа возвращает 160-битное значение.

Процесс шифрования состоит из большого числа итераций, каждая из которых завершается генерацией псевдослучайной функции. Количество пройденных итераций показывает счетчик l. Все они подразделяются на несколько этапов с похожими операциями. На каждом этапе старшие 9 битов

одного из регистров (A, B, C или D) используются в качестве указателя, по которому из таблицы T выбирается значение. Это значение складывается арифметически или поразрядно по модулю 2 (XOR) со следующим регистром (снова один из A, B, C или D). Затем первый выбранный регистр преобразуется циклическим сдвигом вправо на 9 позиций. Далее либо значение второго регистра модифицируется сложением или XOR с содержимым первого (уже сдвинутым) и выполняется переход к следующему этапу, либо этот переход выполняется сразу. После 8 таких этапов значения A, B, C и D складываются (арифметически или XOR) с определенными словами из таблицы S и добавляются в ключевую последовательность y. Завершающий этап итерации заключается в прибавлении к регистрам дополнительных 32-битных значений (n1, n2 или n3, n4). Причем выбор конкретного значения зависит от четности номера данной итерации.

При разработке этого алгоритма главное внимание отводилось следующим свойствам и идеям:

− использование большой (примерно 2 Кбайта) таблицы T, получаемой из большого 160-битного ключа;

− чередование арифметических операций (сложение и побитовый XOR);

− использование внутреннего состояния системы, которое явно не проявляется в потоке данных (значения n1, n2, n3 и n4, которые изменяют регистры в конце каждой итерации);

− использование отличных друг от друга операций в зависимости от этапа итерации и ее номера.

Для шифрования и расшифрования каждого байта текста шифр SEAL требует около четырех машинных тактов. Он работает со скоростью примерно 58 Мбит/с на 32-битном процессоре с тактовой частотой 50 МГц и является одним из самых быстрых шифров.

CRYPTON

CRYPTON — алгоритм симметричного блочного шифрования (размер блока 128 бит, ключ длиной до 256 бит), разработанный южнокорейским криптологом Че Хун Лим (англ. Chae Hoon Lim) из южнокорейской компании Future Systems, с конца 1980-х годов работающая на рынке сетевого обеспечения и защиты информации. Схема работы алгоритма представлена на рис. 54.

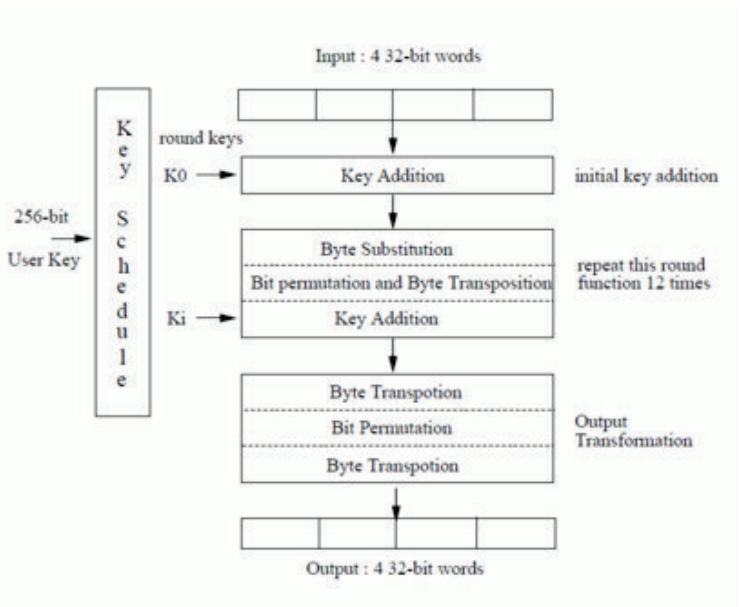

Рисунок 54. Схема алгоритма CRYPTON

Crypton предназначен для шифрования 128-битовых блоков данных. При шифровании используются ключи шифрования нескольких фиксированных размеров — от 0 до 256 бит с кратностью 8 битов.

Структура алгоритма Crypton — структура «Квадрата» — во многом похожа на структуру алгоритма Square, созданного в 1997 году. Криптографические преобразования для алгоритмов с данной структурой могут быть выполнены как над целыми строками и столбцами массива, так и над отдельными его байтами.

Алгоритм Crypton представляет 128-битовый блок шифруемых данных в виде байтового массива 4×4, над которыми в процессе шифрования производится несколько раундов преобразований. В каждом раунде предполагается последовательное выполнение следующих операций:

– табличная замена ;

– линейное преобразование ;

– байтовая перестановка ;

– операция побитового сложения всего массива данных с ключом раунда.

Алгоритм Crypton требует наличие 128 — битового ключа для каждого раунда, а также 128 битового ключа для предварительной операции σ. Расширение ключа происходит в два этапа:

1) на первом этапе происходит формирование восьми расширенных ключей;

2) на втором этапе происходит вычисление ключей раундов из расширенных ключей.

Достоинства алгоритма Crypton:

− алгоритм эффективен на программном и аппаратном уровне благодаря

высокой степени параллельности и использованию очень простых логических операций ANDS/XORS;

− алгоритм не подвержен атакам по времени выполнения и потребляемой мощности;

− хорошая стойкость к существующим атакам;

− возможность распараллеливания операций в процессе шифрования;

− быстрое расширение ключа, быстрое формирование ключей: шифрование со списоком ключей идет намного быстрее чем шифрование с одним блоком, так что это очень эффективно в приложениях, требующих частые замены ключей (например, в хеш-режиме).

− достаточно высока скорость на всех целевых платформах;

− небольшие требования к оперативной памяти и возможность расширения ключа «на лету» позволяют использовать алгоритм Crypton в смарткартах с минимальными ресурсами;

− алгоритм поддерживает дополнительные размеры ключей, помимо тех, что были установлены конкурсом (128, 192, 256 битов).

Недостатки алгоритма Crypton:

− при анализе исходной версии алгоритма, Crypton v0.5, сразу двое экспертов независимо обнаружили класс слабых ключей: таковых оказалось 2 в степени 32 256-битовых ключей;

− была обнаружена атака на шестираундовую версию алгоритма Crypton, похожая на атаку на алгоритм Square. Это было одной из причин появления новой версии алгоритма — Crypton v1.0.

A5

A5 — это поточный алгоритм шифрования, используемый для обеспечения конфиденциальности передаваемых данных между телефоном и базовой станцией в европейской системе мобильной цифровой связи GSM (Groupe Spécial Mobile).

Шифр основан на побитовом сложении по модулю два (булева операция «исключающее или») генерируемой псевдослучайной последовательности и шифруемой информации. В A5 псевдослучайная последовательность реализуется на основе трёх линейных регистров сдвига с обратной связью. Регистры имеют длины 19, 22 и 23 бита соответственно. Сдвигами управляет специальная схема, организующая на каждом шаге смещение как минимум двух регистров, что приводит к их неравномерному движению. Последовательность

формируется путём операции «исключающее или» над выходными битами регистров.

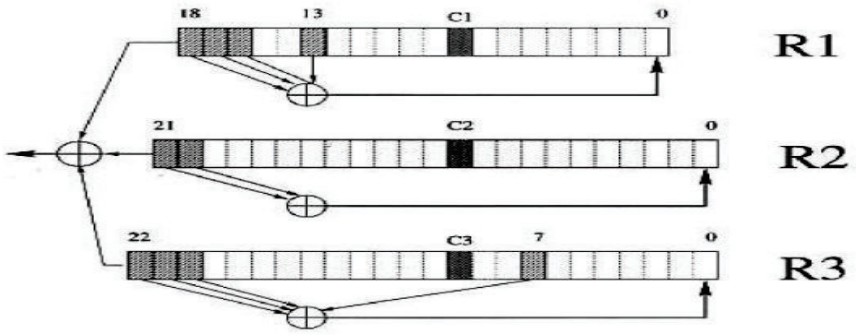

Рисунок 55. Схема A5

Таблица 44. Таблица Кодов

№	Буква	Двоичный код	№	Буква	Двоичный код	№	Буква	Двоичный код
1	А	00000001	12	К	00001100	23	Х	00010111
2	Б	00000010	13	Л	00001101	24	Ц	00011000
3	В	00000011	14	М	00001110	25	Ч	00011001
4	Г	00000100	15	Н	00001111	26	Ш	00011010
5	Д	00000101	16	О	00010000	27	Щ	00011011
6	Е	00000110	17	П	00010001	28	Ъ	00011100
7	Ё	00000111	18	Р	00010010	29	Ы	00011101
8	Ж	00001000	19	С	00010011	30	Ь	00011110
9	З	00001001	20	Т	00010100	31	Э	00011111
10	И	00001010	21	У	00010101	32	Ю	00100000
11	Й	00001011	22	Ф	00010110	33	Я	00100001

Задача: Имея зашифрованное сообщение и ключ, получить исходное сообщение.

Вариант 1

Шифротекст	A9 5 D BC 40
A	100101
B	0111101
C	11001010011

Вариант 2

Шифротекст	41 56 48 5F
A	111010
B	0111100
C	11010101110

Вариант 3

Шифротекст	0C 95 08 97
A	111110
B	0011010
C	11001010100

Вариант 4

Шифротекст	D1 59 CD 59
A	110010
B	1110010
C	11110001101

Вариант 5

Шифротекст	0B 9F 1D 8D
A	001011
B	1100101
C	11100011011

Вариант 6

Шифротекст	F9 0D E7 05
A	000111
B	1110011
C	00001110001

Вариант 7

Шифротекст	5C 41 5C 5F
A	010100
B	1100101
C	11000110101

Вариант 8

Шифротекст	E8 41 ED 49
A	110011
B	1000001
C	10101110001

Вариант 9

Шифротекст	DD AC C2 A9
A	111010
B	0100101
C	01110011010

Вариант 10

Шифротекст	58 BB 4E B3
A	010110
B	1010011
C	10100011011